Rider Weed Deep Residual Network-Based Incremental Model for Text Classification Using Multidimensional Features and MapReduce Framework

Hemn Barzan Abdalla (✉ habdalla@kean.edu)

Kean University

Research Article

Keywords: Text classification, Deep Residual network, MapReduce model, Dynamic learning, Fuzzy theory

Posted Date: September 23rd, 2021

DOI: <https://doi.org/10.21203/rs.3.rs-926862/v1>

License: © ⓘ This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Rider Weed Deep Residual Network-Based Incremental Model for Text Classification Using Multidimensional Features and MapReduce Framework

Hemn Barzan Abdalla

Wenzhou-Kean University, Wenzhou-China

***Corresponding Author: Hemn Barzan Abdalla. Email: habdalla@kean.edu**

Abstract: The increasing demand for information and rapid growth of big data has dramatically increased textual data. The amount of different kinds of data has led to the overloading of information. For obtaining useful text information, the classification of texts is considered an imperative task. This paper develops a technique for text classification in big data using the MapReduce model. The goal is to design a hybrid optimization algorithm for classifying the text. Here, the pre-processing is done with the steaming process and stop word removal. In addition, the Extraction of imperative features is performed wherein SentiWordNet features, contextual features, and thematic features are generated. Furthermore, the selection of optimal features is performed using Tanimoto similarity. The Tanimoto similarity method estimates the similarity between the features and selects the relevant features with higher feature selection accuracy. After that, a deep residual network is utilized for dynamic text classification. The Adam algorithm trains the deep residual network. In addition, the dynamic learning is performed with the proposed Rider invasive weed optimization (RIWO)-based deep residual network along with fuzzy theory. The proposed RIWO algorithm combines Invasive weed optimization (IWO) and the Rider optimization algorithm (ROA). The method mentioned above is solved under the MapReduce framework. The proposed RIWO-based deep residual network outperformed other techniques with the highest True positive rate (TPR) of 85%, True negative rate (TNR) of 94%, and accuracy of 88.7%.

Keywords: Text classification; Deep Residual network; MapReduce model; Dynamic learning; Fuzzy theory

1. Introduction

The massive demand for big data led to evaluate the source and implication of data. The fundamental opinion of analysis relies on designing a novel framework for studying the data. One of the mathematical models is a similarity measure that is utilized for classifying and clustering the data. Here, the fundamental assessment of common similarity measures is provided. Here, the similarity metrics, like Jaccard [10], Cosine [12], Euclidean distance [9], Extended Jaccard [11], are utilized for evaluating distance or angle amongst the vectors. Here, the similarity measures are categorized as feature content and topology. In topology, the features are organized in a hierarchical model, and the appropriate length of path amongst the feature is required to be evaluated. The measures of features are devised on the basis of evidence wherein the features having elevated frequency are employed to be explicit with elevated information, wherein features with less frequency are adapted with less information. Pair-Wise and ITSIm metrics fit into the class of feature content metrics. Here, the information content measure provides elevated priority to the highest features with a small difference amongst the two data that can lead to improved outcomes. Here, the cosine and Euclidean belong to a class of topological metrics. It is susceptible to loss of information as two similar data offset by the existence of solitary feature having huge weight [4]. The methods, such as clustering and classification, are utilized in text mining-based applications that help to transform massive data into small subsets for increasing computational effectiveness [2].

The text data consist of noisy and irrelevant features that made the learning techniques fail to generate improved accuracies. For removing redundant data, various data mining methods are adapted. Here, the selection and Extraction of features are two methods for classifying the data. The techniques, such as

clustering of text for selecting text data features and classification, are adapted recently. The selection of components is utilized for eliminating the superfluous text features for effectively performing classification and clustering. The previous techniques concentrated more on transforming huge data to small data considering classical distance measures. The reduction of dimensionality minimizes evaluation time and maximizes the efficiency of classification. The recovery of data and text are utilized in detecting synonyms and meanings of data. Authors devise several techniques for performing the classification and clustering process. Here, the clustering is carried out using unsupervised techniques with different class label data [2]. The goal of classifying text is to categorize data into different parts. Here, the goal is to allocate pertinent labels based on the content [3].

The categorization of texts is considered an imperative part of processing the natural language. It is extensively employed in several applications. For instance, most new services require repeatedly arrange huge articles in a single day [13]. The advanced services of mails offer the function to discover either junk mail or mail in an automated manner [14]. Other applications involve analysis of sentiment [15], modeling topic [16], text clustering [32], translation of language [17] and intent detection [18] [3]. The classification of technology assists people filters useful data and poses more implications in real life. The design of text categorization and machine learning has shifted manually to machine [19], [20], [21], [22]. With conventional text data for learning and then categorizing the unspecified text is a basic part. There exist several textualization classification techniques [23]. The goal is to categorize textual data. The categorization outcomes can fulfill an individual's requirements for classifying text and is suitable for rapidly attaining significant data. MapReduce is utilized for handling huge data with unstructured data [5].

The aim is to devise an optimization-driven deep learning technique for classifying the texts using big data considering MapReduce. Initially, the text data undergoes pre-processing for removing unnecessary words from data. Here, the pre-processing is performed using the stop word removal and stemming process. After that, the Extraction of features is performed wherein the SentiWordNet features, Thematic features, and contextual features are obtained. These features are employed in a deep residual network for classifying the texts. Here, the training of the deep residual network is performed with the Adams algorithm. Finally, dynamic learning is carried out wherein the proposed RIWO-based deep residual network is employed for incremental text classification. Here, the fuzzy theory is employed for weight bounding in order to deal with the incremental data. In this process, the training of deep residual network is performed with proposed RIWO, which is devised by combining ROA and IWO algorithm

The key contribution of the paper is:

- **Proposed RIWO-based deep residual network for text classification:** The proposed RIWO-based deep residual network is employed for classifying texts by dynamic learning. Here, the developed RIWO is adapted for deep residual network training. The developed RIWO is devised by combining ROA and IWO algorithms. In addition, the fuzzy theory is employed for handling the dynamic data by performing weight bounding.

The rest of the sections are given as: Section 2 presents the survey of classical text classification techniques. Section 3 described the developed text classification model. Section 4 discusses the results of the developed model with respect to classical techniques, and section 5 presents the conclusion.

2. Motivations

The classical text classification techniques using big data with the issues are explained below. The problems and challenges are considered a motivation for developing a novel text classification technique with big data.

2.1. Literature Review

The eight classical techniques based on text classification using big data along with its issues are described below. Nihar M. Ranjan, Rajesh S. Prasad [1] developed an LFNN-based incremental learning technique based on context-semantic features for classifying the text data. The methods employed a dynamic dataset for classification to learn the model in a dynamic manner. Here, the incremental learning procedure employed Back Propagation Lion (BPLion) Neural Network, wherein fuzzy bounding and Lion Algorithm (L.A.) was employed for selecting the weights. However, the technique failed to classify the sentence precisely. Vinay Kumar Kotte *et al.*[2] devised a similarity function for clustering the feature pattern. The imperative feature was that the distribution of words and their dimensionality reduction were similar. The technique attained dimensionality reduction with improved accuracy. However, the technique failed to utilize membership functions for obtaining clusters. Jiaying Wang *et al.*[3] devised a deep learning technique for classifying the text documents. In addition, a large-scale scope-based convolutional neural network (LSS-CNN) was utilized for categorizing the text. The method effectively computed scope-based data and parallel training for massive datasets. The technique attained improved scalability on big data but failed to attain the utmost accuracy. Venkatanaresbhabu Kuppili *et al.*[4] developed Maxwell–Boltzmann Similarity Measure (MBSM) for classifying the text. Here, the MBSM was derived with feature values from the documents. The MBSM was devised by combining single label K-nearest neighbor’s classification (SLKNN), multi-label KNN (MLKNN), and K-means clustering. However, the technique failed to include clustering techniques and query mining. Cheng Liu and Xiaofang Wang [5] devised a technique for classifying the text using English quality-related text data. Here, the goal was to extract, classify and examine the data from English texts considering cyclic neural networks. At last, the features with sophisticated English texts were generated. In addition, the technique combining attention was devised for improving the issue of label disorder and made the structure more reliable. However, the computation cost tends to be very high. Qi Li *et al.* [6] designed a method for classifying text by solving the misfitting issue by performing angle-pruning tasks from a database. Here, the technique computed each convolutional filter’s efficiency using discriminative power produced at the pooling layer and prunes words obtained from the filter. These features were considered for text classification. However, the technique produced elevated computational complexity. Fatma Bensaid and Adel M. Alimi [7] devised Multi-Objective Automated Negotiation-based Online Feature Selection (MOANOFs) for classifying the texts. Here, the MOANOFs utilized Automated Negotiation and machine learning techniques for improving the performance of classification using ultra-high dimensional datasets. This helped the method to decide which features are the most pertinent. However, the method failed to select features from multi-classification domains. Jiang Met *al.* [8] devised a hybrid text classification model based on softmax regression for classifying text. Here, the deep belief network was utilized for performing the classification of text using learned feature space. However, the technique failed to filter extraneous characters for enhancing system performance.

2.2. Challenges

The issues confronted in the classification of text using big data are given below,

- The MBSM technique attained improved classification of texts. However, the major issue is the adaption of MBSM in big data. However, it can cause various social and environmental issues [4].
- The ReDP-CNN acquired low computational memory and power, but the issue relies on addressing the text classification issue under unconstrained resources [6].
- In [8], the Soft regression model is devised for classifying the text considering the big data. However, this model did not filter inappropriate features that affected the accuracy of classification [8].
- The major issue of MOANOFs is the issues of the online selection of features considering nonlinear domains [7].
- The text data issue relies on improving the accuracy of a classification and enhancing the clustering quality.

3. Proposed RIWO-based deep residual network for text classification in big data

The mission of text classification is to categorize text data into different classes based on certain content. The classification of text is considered an imperative role in processing the natural language. However, text classification is considered a challenging issue due to high dimensional and noisy texts, which is considered a complex process to devise an improved classifier for huge textual data. This research devises a novel hybrid optimization-driven deep learning technique for text classification using big data. Here, the goal is to devise a classifier that employs text data as input and allocates pertinent labels based on the content. At first, the input text data undergoes pre-processing to eliminate noise and artifacts in the text data. Here, the pre-processing is performed with stop word removal and stemming. Once the pre-processed data is obtained, the features, like contextual features, thematic features, and SentiWordNet features, are extracted. Once the features are extracted, the imperative features are chosen with Tanimoto similarity. The Tanimoto similarity method evaluates similarity amongst features and chooses the relevant features having high feature selection accuracy. Once the features are selected, a deep residual network [26] is used for dynamic text classification. The deep residual network is trained using the Adam algorithm [11], [33], [34]. In addition, dynamic learning is performed using the proposed RIWO algorithm along with the fuzzy theory. The proposed RIWO algorithm is the integration of IWO [27] and ROA [28]. Figure 1 shows a schematic view of the text classification model with big data using the proposed RIWO method.

Figure 1. Schematic view of text classification model with big data using proposed RIWO method

Assume input text data expressed as B with various attributes and is expressed as

$$B = \{B_{d,e}\}; (1 \leq d \leq D)(1 \leq e \leq E) \quad (1)$$

where $B_{d,e}$ refers to text data contained in the database B with e^{th} attribute in d^{th} data. Here, D data points are employed using E attributes for each data. The other step is to eliminate artifacts and noise present in the data.

The data $B_{d,e}$ in a database is split into a specific number, which is equivalent to mappers present in the MapReduce model. The partitioned data is given by,

$$B_{d,e} = \{D_q\}; 1 \leq q \leq N \quad (2)$$

where N symbolize total mappers. Assume N mappers in MapReduce be expressed as

$$M = \{M_1, M_2, \dots, M_q, \dots, M_N\}; 1 \leq q \leq N \quad (3)$$

Thus, input to q^{th} mapper is formulated as,

$$D_q = \{d_{r,l}\}; 1 \leq r \leq m_q; 1 \leq l \leq n \quad (4)$$

where, $d_{r,l}$ symbolize split data given to q^{th} mapper to process, and m_q indicate data in q^{th} mapper.

3.1 Pre-processing

The partitioned data $d_{r,l}$ from the text dataset is fed to pre-processing phase to remove the superfluous words by removing stop words and stemming. Here, pre-processing is known as an important process to arrange various data in a smooth manner for offering effective outcomes—the pre-processing assists in explaining processing for obtaining improved representations. The dataset consists of unnecessary phrases and words that influence the process. Thus, pre-processing is an important process for removing inconsistent words from the dataset. Initially, the text data are accumulated in the relational dataset, and all reviews are divided into sentences and bags of the sentence. Thus, the elimination of stop words is carried out to maximize the performance of the text classification model. Here, the stemming and stop word removal are done to refine data.

(i) Stop word removal

It is a process to remove words with less representative value for data. Some of the instances of non-representative words are pronouns and articles. While evaluating data, there exist few words which are not valuable to text content. Thus, removing such redundant words is imperative, and this procedure is termed stop word removal [29]. The continual happening of words, like articles, conjunctions, and prepositions, such as the, is, a, an, and, when, but some high-frequency words are adapted as stop words. In addition, the removal of stop words is the most imperative technique, which is utilized to remove such redundant words using vocabulary as the size of vector space does not offer any meaning, and thus it is employed as less imperative. The stop words indicate word, which does not hold any data. It is a process to eliminate stop words from a huge set of reviews. The elimination of stop word is used for saving huge-space and for performing processing faster in order to attain an effective process.

(ii) Stemming

The stemming procedure is utilized to convert words to stem. In massive data, several words are utilized which convey a similar meaning. Thus, the critical method utilized to minimize words to root is termed stemming. Stemming is a method of linguistic normalization wherein insignificant words are reduced to general means. Moreover, stemming is the procedure to retrieve information for describing the mechanism

for removing redundant words to their root form and word stem. For instance, if a word starts from connections, connection, connecting, connected, the word is reduced as connect [29].

$$M_l = \{Q_i, 1 \leq i \leq P_k\} \quad (5)$$

where P_k symbolize total words present in text data from the database.

The pre-processed outcome generated from pre-processing is expressed as, B' which is subjected as an input to feature extraction phase

3.2 Acquisition of features for producing highly pertinent features

It describes an imperative feature produced with input review, and the implication of feature extraction is to produce pertinent features that facilitate improved classification of text. Moreover, the obstruction of data is reduced as text data is expressed as a minimized feature set. Thus, the pre-processed partitioned data $d_{r,l}$ is fed to feature Extraction, wherein mining of SentiWordNet features, contextual features, and thematic features is performed.

3.2.1. Extraction of SentiWordNet features

From pre-processed partitioned data $d_{r,l}$, the SentiWordNet features are utilized by removing keywords from reviews employing each review. Here, the SentiWordNet [31] is employed as a lexical resource to extract the SentiWordNet features. The SentiWordNet assigns each text of WordNet considering three numerical sentiment scores, like positive score, negative score, and neutral score. Here, different words indicate different polarity that indicates various word senses. The SentiWordNet consists of different linguistic features that consist of verbs, adverbs, adjectives, and n-gram features. SentiWordNet is a process to evaluate the score of a specific word using text data. Here, the SentiWordNet is employed to determine the polarity of offered review. Thus, the SentiWordNet is employed for discovering positivity and negativity. Hence, SentiWordNet is modeled as F_1 .

3.2.2. Extraction of contextual features

From pre-processed partitioned data $d_{r,l}$, the context-based features [1] are generated that describes relevant words by dividing them through non-relevant reviews for effective classification. It requires discovering key terms that attain context terms and semantic meaning, which establishes a pertinent context. The key term is considered as a preliminary indicator for relevant review wherein context terms act as a validator that evaluates if determined key term as an indicator. Here, the training dataset contains keywords with pertinent words. The context-based features assist in selecting the pertinent and non-relevant reviews.

Consider N representing a training dataset that poses N_{rel} as relevant reviews and N_{non_rel} as non-relevant reviews. In this method, assume x_s represent key term and x_c indicate context term.

- Detection of Key terms:

Consider L symbolize language model, that employs each term, and a metric is expressed as,

$$C = \frac{L_{rel}}{L_{on_rel}} \quad (6)$$

where L_{rel} symbolize language model for N_{rel} and $L_{non-rel}$ signifies language model for N_{non_rel} .

- *Discovery of Context term:*

After discovering key terms, the technique begins the process of context term discovery, which is similar to first to detect each term in a separate manner. The steps employed in determining the context term are given as,

i) Compute all instances of the key term employed amid relevant and non-relevant reviews, N_{rel} and N_{non_rel} .

ii) By employing sliding window size S , the conditions amidst k_s are mined as context terms. Hence, the size of the window S is employed as a context span.

iii) The pertinent terms generated are employed as a text, modeled as d_r , and non-relevant is denoted as d_{nr} . The set of pertinent text is modeled as R_{d_r} , and the non-relevant set is referred as R_{d_nr} .

iv) Thereafter, the score is evaluated for each distinctive term expressed as,

$$C(x_C) = |L_{R_{d_r}}(x_C) - L_{R_{d_nr}}(x_C)|_S \quad (7)$$

where $L_{R_{d_r}}(x_C)$ symbolize language model for an excerpt with relevant review set. The term $L_{R_{d_nr}}(x_C)$ indicates a language model for an excerpt with a non-relevant review set and S represents the size of the window. If the measure is afar definite threshold, then that score is adapted as a context term x_s . Thus, generated context-based features are modeled as F_2 .

3.2.3. Extraction of thematic features

Here, the pre-processed partitioned data $d_{r,l}$ is given as input to find thematic features. Here, the count of the thematic word [30] in a sentence is the imperative feature as the words that occur frequently are most probably connected to the topic in a data. Thematic words are words that grab key topics defined in a provided document. In thematic features, the top most 10 frequent words are employed as a thematic. Thus, the thematic feature F_3 is modeled as,

$$F_1 = \frac{T}{\sum_{t=0}^{\ell} T_t} \quad (8)$$

Here, T express count of thematic words in a sentence of data, and it is expressed as $\{T_1, T_2, \dots, T_\ell\}$.

Thus, the feature vector considering the contextual, thematic, and SentiWordNet features are expressed as,

$$F = \{F_1, F_2, F_3\} \quad (9)$$

where F_1 symbolize SentiWordNet features, F_2 signifies contextual features, and F_3 refers to thematic features.

3.3. Feature selection using Tanimoto similarity

The selection of imperative features from the extracted features F is made using Tanimoto similarity. The Tanimoto similarity computes similarity amidst features and selects pertinent features having elevated feature selection accuracy. Here, the Tanimoto similarity is expressed as,

$$S = \frac{\sum_{w=1}^m y_w z_w}{\sum_{w=1}^m y_w^2 + \sum_{w=1}^m z_w^2 - \sum_{w=1}^m y_w z_w} \quad (10)$$

where, S indicates Tanimoto measure, y_w and z_w represents features. The selected features are expressed as R .

The produced feature selection output obtained from the mapper is given as input to the reducer U . The classification of texts is performed on reducer using the selected features and briefly illustrated below.

3.4. Classification of texts with Adam-based deep residual network

Here, the classification of text is performed with an Adam-based deep residual network using selected features R . The classification of text data assists in standardizing the infrastructure and makes the search simpler and more pertinent. In addition, the classification enhances the experience of the user and simplifies navigation. In addition, text classification helps to solve huge business issues in real-time, like social media, e-mails, which can speed work and take less time for processing. The deep residual network is more effective in the case of the count of attributes and computation. This network is capable of building deep representations at each layer. In addition, it can manage advanced deep learning tasks. The architecture of the deep residual network and training with the Adams algorithm is described below.

3.4.1. Architecture of Deep residual network

Here, a deep residual network [26] is employed to make an effectual decision in which the classification of text is performed. The DRN comprises different layers, namely residual blocks, convolutional (Conv) layers, linear classifier, and average pooling layers. Figure 2 presents the structural design of the deep residual network.

-Convolutional (Conv) layer:

The two-dimensional Conv layer is utilized to reduce free attributes in training, and it offers reimbursement for allocating weight. The cover layer process the input image with the sequence of filter

known as kernel using a local connection. The cover layer utilizes a mathematical process for sliding the filter with the input matrix and computes the dot product of the kernel. The evaluation process of Conv layer is represented as,

$$B2d(O) = \sum_{a=0}^{E-1} \sum_{s=0}^{E-1} X_{a,s} \bullet O_{(u+a),(v+s)} \quad (11)$$

$$B1d(O) = \sum_{Z=0}^{C_{in}-1} G_Z * O \quad (12)$$

where, O express CNN feature of the input image, u and v refers recording coordinates, G signifies $E \times E$ kernel matrix, and is termed as a learnable parameter, a and s is position index of the kernel matrix. Hence, G_Z express the size of the kernel for Z^{th} input neuron, and $*$ express cross-correlation operator.

-Pooling layer: This layer is associated with the Conv layer and is especially utilized for reducing the spatial size of the feature map. Hence, the average pooling is selected to function on each slice and depth of the feature map.

$$a_{out} = \frac{a_{in} - Z_a}{\lambda} + 1 \quad (13)$$

$$s_{out} = \frac{s_{in} - Z_s}{\lambda} + 1 \quad (14)$$

where a_{in} symbolize input matrix width, s_{in} signifies the height of input matrix, a_{out} and s_{out} represents the respective value of output. In addition, Z_a and Z_s symbolize the width and height of kernel size.

-Activation function: The nonlinear activation function is adapted for learning nonlinear and complicated features in such a way that it is utilized to improve the non-linearity of extracted features. Rectified linear unit (ReLU) is utilized for processing data. The ReLU function is formulated as,

$$ReLU(O) = \begin{cases} 0 & ; K < 0 \\ K & ; K \geq 0 \end{cases} \quad (15)$$

Here, K symbolize a feature.

-Batch normalization: Here, the training set is divided into various small sets known as mini-batches to train the model. It attains a balance between evaluation and convergence complexity. Here, the input layers are normalized by scaling activations to maximize reliability and training speed.

-Residual blocks: It indicates shortcut connection amongst Conv layers. The input is unswervingly allocated to output only if input and output are of equal size.

$$\kappa = \Re(O) + O \quad (16)$$

$$\kappa = \Re(O) + \lambda_M O \quad (17)$$

Here, O and κ signifies input and output residual blocks, κ symbolizes mapping relation, and λ_M expresses dimension matching factor, and $\Re(\cdot)$ signifies activation function.

-Linear classifier: After completion of Conv layer, linear classifier performs a procedure to discover noisy pixels using input features. It is the combination of softmax function and a fully connected layer.

$$\kappa = \hat{\lambda}\kappa + \nu \quad (18)$$

Here, $\hat{\lambda}$ express weight matrix, and ν represent bias—figure 2 indicates structural design of the deep residual network. Here, the output is represented as κ that assists in classifying the texts.

Figure 2. Structural design of deep residual network

3.4.2. Training of Deep residual network with Adams algorithm

The deep residual network training is performed with the Adams technique that assists in discovering the best weights for tuning the deep residual network for classifying text. The optimal weights are produced with the Adams method, which assists in tuning the deep residual network for generating the best results. Adam [11] represents a first-order stochastic gradient-based optimization that is extensively adapted to a fitness function that changes for attributes. The major implication of the method is computational efficiency and fewer memory needs. Moreover, the problems associated with the non-stationary objectives and the subsistence of noisy gradients are handled in an effective manner. In addition, Adam contains the following benefits. Here, the magnitudes of updated parameters are invariant in contrast to rescaling of gradient, and step size is handled with a hyperparameter that works with sparse gradients. In addition, the Adams is effective in performing step size annealing. The classification of text employs a deep residual network for texts. The steps of Adam are given as:

Step 1: Initialization

The foremost step represents bias corrections initialization wherein $\hat{q}_l$ signifies corrected bias of first moment estimate and $\hat{m}_l$ represents corrected bias of second moment estimate.

Step 2: Discovery of error

The error of bias is computed to choose the optimum weight for training the deep residual network. Here, the error is termed as an error function that leads to an optimal global solution. The function is termed as a minimization function and is expressed as,

$$Err = \frac{1}{f} \sum_{l=1}^f (O_l - \kappa)^2 \quad (19)$$

Where f signifies total data, κ symbolizes output generated with Deep residual network classifier, O_l indicates expected value.

Step 3: Discovery of updated bias

Adam is used to improving convergence behavior and optimization. This technique generates smooth variation with effectual computational efficiency and lower memory requirements. As per Adam [11], the bias is expressed as,

$$\theta_l = \theta_{l-1} - \frac{\alpha \hat{q}_l}{\sqrt{\hat{m}_l + \varepsilon}} \quad (20)$$

where α refers step size, $\hat{q}_l$ express corrected bias, $\hat{m}_l$ indicate bias-corrected second-moment estimate, ε represent constant, θ_{l-1} signifies parameter at a prior time instant $(l-1)$. The corrected bias of the first-order moment is expressed as,

$$\hat{q}_l = \frac{q_l}{(1 - \eta_1^l)} \quad (21)$$

$$\hat{q}_l = \eta_1 q_{l-1} + (1 - \eta_1) G_l^1 \quad (22)$$

The corrected bias of second order moment is represented as,

$$\hat{m}_l = \frac{m_l}{(1 - \eta_2^l)} \quad (23)$$

$$\hat{m}_l = \eta_2 m_{l-1} + (1 - \eta_2) H_l^2 \quad (24)$$

$$\text{where, } H_l = \nabla_{\theta} \text{loss}(\theta_{l-1}) \quad (25)$$

Step 4: Determination of best solution: The best solution is determined with error, and a solution having a better solution is employed for classifying text.

Step 5: Termination: The optimum weights are produced repeatedly till utmost iterations are attained. Table 1 describes the pseudocode of the Adams technique.

Table 1. Pseudocode of Adams algorithm

Input: $\hat{q}_l$: corrected bias of first moment estimate and $\hat{m}_l$: corrected bias of second moment estimate
Output: Resulting parameters
Require α step size
Require $\eta_1, \eta_2 \in [0,1)$: Exponential rate decay
Initialize the first-moment vector
Initialize second-moment vector
Initialize time step
While θ_l not converged, do
$l \leftarrow l + 1$
Get gradients
Update first biased moment estimate using equation (22)
Update second biased moment estimate using equation (24)
Evaluate corrected bias first-moment estimate
Evaluate corrected bias second-moment estimate
Update parameters
End while
Return θ_l

3.5. Dynamic learning with proposed RIWO-based deep residual network

For incremental data B , dynamic learning is done using the proposed RIWO-based deep residual network. Here, the assessment of incremental learning with the developed RIWO-based deep residual network is done to achieve effective text classification with the dynamic data. The deep residual network is trained with developed RIWO for generating optimum weights. The developed RIWO is generated by integrating ROA and IWO for acquiring effective dynamic text classification.

3.5.1. Architecture of deep residual network

The model of the deep residual network is already explained in section 3.4.

3.5.2. training of deep residual network with proposed RIWO

The training of deep residual networks is performed with developed RIWO, which is devised by integrating IWO and ROA. Here, the ROA [28] is motivated by the behavior of rider groups, which travel to attain a common target position to turn out to be a winner. In this model, the riders are chosen from the total riders of each group. Hence, it is concluded that this method produces enhanced accuracy of classification. Furthermore, the ROA is effective and follows the steps of fictional computing for addressing optimization problems but contains less convergence. IWO [27] is motivated by colonizing characteristics of weed plants. The technique provided a fast rate of convergence and elevated the accuracy. Hence, the integration of IWO and ROA is carried out to enhance complete algorithmic performance. The steps present in the method are expressed as:

Step 1) Initialization of population

The preliminary step is algorithm initialization, which is performed using four-rider groups provided by A and represented as,

$$A = \{A_1, A_2, \dots, A_\mu, \dots, A_g\} \quad (26)$$

where, A_μ signifies μ^{th} rider, and $\mathcal{Q}$ is total riders.

Step 2) Determination of error:

The computation of error is already described in equation (19).

Step 3) Update riders position:

The rider position in each set is updated for determining the leader. Thus, the rider updates position using a feature of each rider is defined below. The update position of each rider is expressed as,

As per ROA [28], the update overtaker position is used in the update to increase the rate of success by determining the position of overtaker and is represented as,

$$A_{n+1}^o(g, h) = A_n^o(g, h) + [\hat{\partial}_n^*(g) * A^L(L, h)] \quad (27)$$

where $\hat{\partial}_n^*(g)$ signifies direction indicator.

The attacker contains a propensity to grab the position of leaders and given by,

$$A_{n+1}^a(g, u) = A^L(L, u) + [Cos K_{g,u}^n * A^L(L, u)] + r_g^n \quad (28)$$

The bypass riders contain a familiar path, and its update is expressed as,

$$A_{n+1}^b(g, u) = \lambda [A_n(\chi, u) * \delta(u) + A_n(\xi, u) * [1 - \delta(u)]] \quad (29)$$

where λ symbolize random number, χ signifies arbitrary number amongst 1 to P , ξ denote an arbitrary number in 1 to P and δ express arbitrary number between 0 and 1.

The follower poses a propensity to update position using leading rider position to attain target and given by,

$$A_{n+1}^F(g, h) = A^L(L, h) + [Cos(K_{g,h}^n * A^L(L, h) * r_g^n)] \quad (30)$$

where, h is coordinate selector, A^L indicate leading rider position, L represent leading rider index, $K_{g,h}^n$ represent steering angle of g^{th} rider in h^{th} coordinate, and r_g^n is the distance.

$$A_{n+1}^F(g, h) = A^L(L, h) [1 + Cos(K_{g,h}^n * r_g^n)] \quad (31)$$

The IWO assists in generating the best solutions. Hence, as per IWO [27], the equation is represented as,

$$A_{n+1}^F(g, h) = \sigma(n)A_n^F + A_{best} - A_n^F \quad (32)$$

where A_{n+1}^F symbolize new weed position in iteration $n + 1$, A_n^F signifies current weed position A_{best} refers the best weed found in the whole population and $\sigma(n)$ represents current standard deviation.

$$A_{best} = A_{n+1}^F(g, h) - \sigma(n)A_n^F + A_n^F \quad (33)$$

Substitute equation (33) in equation (31),

$$A_{n+1}^F(g, h) = \left(A_{n+1}^F(g, h) - \sigma(n)A_n^F + A_n^F \right) \left[1 + \cos(K_{g,h}^n) * r_g^n \right] \quad (34)$$

$$A_{n+1}^F(g, h) = A_{n+1}^F(g, h) \left[1 + \cos(K_{g,h}^n) * r_g^n \right] - \left(\sigma(n)A_n^F + A_n^F \right) \left[1 + \cos(K_{g,h}^n) * r_g^n \right] \quad (35)$$

$$A_{n+1}^F(g, h) - A_{n+1}^F(g, h) \left[1 + \cos(K_{g,h}^n) * r_g^n \right] = \left[A_n^F - \sigma(n)A_n^F \right] \left[1 + \cos(K_{g,h}^n) * r_g^n \right] \quad (36)$$

$$A_{n+1}^F(g, h) \left[1 - 1 - \cos(K_{g,h}^n) * r_g^n \right] = \left[A_n^F - \sigma(n)A_n^F \right] \left[1 + \cos(K_{g,h}^n) * r_g^n \right] \quad (37)$$

$$A_{n+1}^F(g, h) \left[-\cos(K_{g,h}^n) * r_g^n \right] = \left[A_n^F - \sigma(n)A_n^F \right] \left[1 + \cos(K_{g,h}^n) * r_g^n \right] \quad (38)$$

The final update equation of the proposed RIWO is expressed as,

$$A_{n+1}^F(g, h) = - \frac{(A_n^F - A_n^F \sigma(n)) (1 + \cos(K_{g,h}^n) * r_g^n)}{\cos(K_{g,h}^n) * r_g^n} \quad (39)$$

Step 4) Re-evaluation of the error:

After completing the process of update, the error of each rider is computed. Here, the position of the rider who is in the leading position is replaced using the position of the new rider generated such that the error of the new rider is less.

Step 5) Update of Rider parameter:

The rider attribute update is imperative to determine an effectual optimal solution using error.

Step 6) Riding Off time:

The steps are iterated repeatedly till time attains off time N_{OFF} , in which the leader is determined. The Pseudocode of developed RIWO is illustrated in table 2.

Table2. Pseudocode of developed RIWO

Input: A : Arbitrary rider position, n : iteration, $n_{\max}$: maximum iteration
Output: Leader A^L
Begin
Initialize solutions set
Initialize algorithmic parameter

Discover error using equation (19)
While $n < N_{OFF}$
For $\nu = 1$ to P
Update bypass position with equation (29)
Update follower position with equation (39)
Update overtaker position with equation (27)
Update attacker position with equation (28)
Rank riders using error with equation (19)
Choose the rider with minimal error
Update steering angle, gear, accelerator, and brake
Return A^L
$n = n + 1$
End for
End while
End

Hence, the output produced from the developed RIWO-based deep residual network is κ , which helps to classify the text data considering dynamic learning that helps to classify the dynamic data. Here, fuzzy bounding is employed for remodeling the classifier if an error of previous data with respect to present data is high.

3.5.3 Fuzzy theory

Whenever incremental data is added to the model, the error is evaluated, and weights are updated without using the previous weights. If the error evaluated by the present instance is less than the error of the previous instance then, the weights are updated based on the proposed RIWO algorithm; else, if the error computed by the current instance is more than the error of the previous instance, then the classifiers are remodeled by setting a boundary for weight using fuzzy theory [1] and then, choose optimal weight using proposed RIWO algorithm. On arrival of data d_{i+1} , the error e_{i+1} will be computed, and that will be compared with that of previous data d_i . If $e_i < e_{i+1}$ then prediction with training based on RIWO is made else the fuzzy bounding based learning will be done by bounding the weights, which is given as,

$$\omega^B = \omega^t \pm F^s \quad (40)$$

where, ω^t is weight at current iteration, and F^s signifies fuzzy score. For the dynamic data, extract the features $\{F\}$. Here, the membership degree is given as,

$$Membership\ Degree = \|\omega^{t-2} - \omega^{t-1}\| \quad (41)$$

where ω^{t-2} represent weights at iteration $t - 2$ and ω^{t-1} signifies weights at iteration $t - 1$. At last, when the highest iteration is attained, the process is stopped.

4. Results and Discussion

The competence of the technique is evaluated by analyzing the techniques with various measures like TPR, TNR, and accuracy. The assessment is done by considering mappers=3, mappers=4, and by varying the chunk size.

4.1 Experimental setup

The execution of the developed model is performed in PYTHON with windows 10 OS, Intel processor, 4GB RAM. Here, the analysis is performed by considering the NSL-KDD dataset.

4.2. Dataset description

The dataset adapted for text classification involves Reuter and 20 Newsgroups database and is explained below.

4.2.1. 20 Newsgroups database:

The 20 Newsgroups data set [24] is contributed by Ken Lang for the newsreader to extract the Netnews. The dataset is established by collecting 20,000 newsgroup data, which is split amongst 20 different newsgroups. The database is popular for analyzing text applications for handling machine learning methods, like clustering and classification of text. The dataset is organized into 20 different newsgroups, wherein each indicates different topics.

4.2.2. Reuter database:

The Reuters-21578 Text Categorization Collection Data Set [25] is contributed by David D. Lewis. The dataset comprises documents that occur on Reuters newswires in 1987. The documents are arranged and indexed based on categories. The count of instances of the dataset is 21578 has five attributes. The count of websites attained by dataset is 163417.

4.3 Evaluation metrics

The efficiency of the developed model is examined by adopting measures like accuracy, TPR, and TNR.

4.3.1. accuracy:

It is described as the measure of data that is precisely preserved and is expressed as,

$$Acc = \frac{P + Q}{P + Q + H + F} \quad (42)$$

where P signifies true positive, Q symbolizes true negative, H denote true false positive, and F is a false negative.

4.3.2. TPR

TPR refers ratio of the count of true positives with respect to the total number of positives.

$$TPR = \frac{P}{P + H} \quad (43)$$

where, P refers true positives, H is the false negatives.

4.3.3. TNR

The TNR refers to the ratio of negatives that are correctly detected.

$$TNR = \frac{Q}{Q+F} \quad (44)$$

where Q is true negative and F signifies false positive.

4.4. Comparative methods

The analysis of the proposed technique is evaluated by evaluating methods with classical techniques, like LSS-CNN [3], RNN [5], SLKNN+MLKNN [4], BPLion+LFNN [1], and proposed RIWO-based deep residual network.

4.5. Comparative analysis

The assessment of the proposed technique is performed by adopting certain measures, like accuracy, TPR, and TNR. Here, the analysis is performed by considering two datasets, namely Reuter dataset and the 20 Newsgroup dataset. In addition, the assessment of techniques is performed by considering the mapper size=3 and 4.

4.6.1. Analysis with Reuter dataset

The assessment of techniques with the Reuter dataset considering TPR, TNR, and accuracy parameters is described. The assessment is done with mapper=3 and mapper=4 by varying the chunk size.

a) Assessment with mapper=3

Figure 3. Assessment of techniques considering Reuter dataset with mapper=3 using a) TPR b) TNR c) Accuracy

Figure 3 presents an assessment of techniques with accuracy, TPR and TNR measure considering with Reuter dataset with mapper=3. The assessment of techniques with TPR measure is depicted in figure 3a). For chunk size=3, the TPR evaluated by LSS-CNN, RNN, SLKNN+MLKNN, BPLion+LFNN, and proposed RIWO-based deep residual network are 0.747, 0.757, 0.771, 0.790, and 0.803. Likewise, for

chunk size=6, the TPR was evaluated by LSS-CNN, RNN, SLKNN+MLKNN, BPLion+LFNN, and proposed RIWO-based deep residual network are 0.800, 0.812, 0.818, 0.819, and 0.830. The assessment of techniques with TNR measure is depicted in figure 3b). For chunk size=3, the TNR was evaluated by LSS-CNN, RNN, SLKNN+MLKNN, BPLion+LFNN, and proposed RIWO-based deep residual network are 0.831, 0.842, 0.853, 0.886, and 0.913. Likewise, for chunk size=6, the TNR was evaluated by LSS-CNN, RNN, SLKNN+MLKNN, BPLion+LFNN, and proposed RIWO-based deep residual network are 0.846, 0.850, 0.865, 0.896, and 0.925. The assessment of the method with accuracy measure is depicted in figure 3c). For chunk size=3, the accuracy was evaluated by LSS-CNN, RNN, SLKNN+MLKNN, BPLion+LFNN, and proposed RIWO-based deep residual network are 0.820, 0.831, 0.835, 0.857, and 0.870. Likewise, for chunk size=6, the accuracy was evaluated by LSS-CNN, RNN, SLKNN+MLKNN, BPLion+LFNN, and proposed RIWO-based deep residual network is 0.824, 0.839, 0.849, 0.863, and 0.880. The performance improvement of LSS-CNN, RNN, SLKNN+MLKNN, BPLion+LFNN with respect to the proposed RIWO-based deep residual network considering accuracy are 6.363%, 4.659%, 3.522%, and 1.931%.

b) Assessment with mapper=4

Figure 4. Assessment of techniques considering Reuter dataset with mapper=4 using a) TPR b) TNR c) Accuracy

The assessment of techniques with accuracy, TPR, and TNR measure considering with Reuter dataset using mapper=4 is described in figure 4. The assessment of techniques with TPR is displayed in figure 4a). For chunk size=3, the TPR evaluated by LSS-CNN is 0.754, RNN is 0.768, SLKNN+MLKNN is 0.792, BPLion+LFNN is 0.810, and the proposed RIWO-based deep residual network is 0.828. Likewise, for chunk size=6, the TPR evaluated by LSS-CNN is 0.810, RNN is 0.820, SLKNN+MLKNN is 0.824, BPLion+LFNN is 0.826, and the proposed RIWO-based deep residual network is 0.850. The assessment of techniques with TNR measure is depicted in figure 4b). For chunk size=3, the TNR evaluated by LSS-CNN is 0.839, RNN is 0.860, SLKNN+MLKNN is 0.863, BPLion+LFNN is 0.896, and the proposed RIWO-based deep residual network is 0.925. Likewise, for chunk size=6, the TNR evaluated by LSS-CNN is 0.855, RNN is 0.856, SLKNN+MLKNN is 0.876, BPLion+LFNN is 0.900, and the proposed RIWO-based deep residual network is 0.940. The assessment of the method with accuracy measure is displayed in figure 4c). For chunk size=3, the accuracy evaluated by LSS-CNN is 0.837, RNN is 0.843, SLKNN+MLKNN is

0.846, BPLion+LFNN is 0.862, and the proposed RIWO-based deep residual network is 0.880. Likewise, for chunk size=6, the accuracy evaluated by LSS-CNN is 0.833, RNN is 0.849, SLKNN+MLKNN is 0.852, BPLion+LFNN is 0.868, and the proposed RIWO-based deep residual network is 0.887. The performance improvement of LSS-CNN, RNN, SLKNN+MLKNN, BPLion+LFNN with respect to the proposed RIWO-based deep residual network considering accuracy are 6.087%, 4.284%, 3.945%, and 2.142%.

4.6.2. analysis with 20 Newsgroup dataset

The assessment of techniques using 20 Newsgroup datasets with TPR, TNR, and accuracy parameters is elaborated. The assessment is done with mapper=3 and mapper=4 by altering the chunk size.

a) Assessment with mapper=3

Figure 5. Assessment of techniques considering 20 Newsgroup dataset with mapper=3 using a) TPR b) TNR c) Accuracy

Figure 5 presents an assessment of techniques with accuracy, TPR and TNR measure considering with 20 Newsgroup dataset with mapper=3. The assessment of techniques with TPR measure is depicted in figure 5a). For chunk size=3, the maximal TPR of 0.834 is evaluated by the proposed RIWO-based deep residual network, while TPR is evaluated by LSS-CNN, RNN, SLKNN+MLKNN, and BPLion+LFNN are 0.708, 0.759, 0.780, and 0.812. Likewise, for chunk size=6, the highest TPR of 0.840 is evaluated by the proposed RIWO-based deep residual network, while the TPR is evaluated by LSS-CNN, RNN, SLKNN+MLKNN, and BPLion+LFNN are 0.796, 0.815, 0.818, and 0.829. The assessment of techniques with TNR measure is depicted in figure 5b). For chunk size=3, the TNR computed by the proposed RIWO-based deep residual network is 0.862, while TNR is evaluated by LSS-CNN, RNN, SLKNN+MLKNN, BPLion+LFNN are 0.832, 0.839, 0.843, 0.851. Likewise, for chunk size=6, the TNR evaluated by the proposed RIWO-based deep residual network is 0.879 while TNR computed by LSS-CNN, RNN, SLKNN+MLKNN, BPLion+LFNN are 0.833, 0.840, 0.849, 0.854. The assessment of the method with accuracy measure is depicted in figure 5c). For chunk size=3, the accuracy evaluated by the proposed RIWO-based deep residual network is 0.850, while accuracy is computed by LSS-CNN, RNN, SLKNN+MLKNN, BPLion+LFNN are 0.821, 0.822, 0.833, 0.843. The performance improvement of LSS-CNN, RNN, SLKNN+MLKNN, BPLion+LFNN with respect to the proposed RIWO-based deep residual network considering accuracy are 3.411%, 3.294%, 2%, and 0.823%. Likewise, for chunk size=6, the

accuracy evaluated by the proposed RIWO-based deep residual network is 0.859, while accuracy is computed by LSS-CNN, RNN, SLKNN+MLKNN, and BPLion+LFNN are 0.824, 0.830, 0.839, and 0.856.

b) Assessment with mapper=4

Figure 6. Assessment of techniques considering 20 Newsgroup dataset with mapper=4 using a) TPR b) TNR c) Accuracy

Figure 6 presents an assessment of techniques with accuracy, TPR, and TNR measure considering with 20 Newsgroup dataset with mapper=4. The assessment of techniques with TPR measure is depicted in figure 6a). For chunk size=3, the TPR evaluated by LSS-CNN, RNN, SLKNN+MLKNN, BPLion+LFNN, and proposed RIWO-based deep residual network are 0.721, 0.769, 0.798, 0.810, and 0.845. Likewise, for chunk size=6, the TPR was evaluated by LSS-CNN, RNN, SLKNN+MLKNN, BPLion+LFNN, and proposed RIWO-based deep residual network are 0.810, 0.827, 0.836, 0.849, and 0.859. The assessment of techniques with TNR measure is depicted in figure 6b). For chunk size=3, the TNR was evaluated by LSS-CNN, RNN, SLKNN+MLKNN, BPLion+LFNN, and proposed RIWO-based deep residual networks are 0.831, 0.831, 0.842, 0.867, and 0.870. Likewise, for chunk size=6, the TNR was evaluated by LSS-CNN, RNN, SLKNN+MLKNN, BPLion+LFNN, and proposed RIWO-based deep residual network are 0.836, 0.839, 0.851, 0.871, and 0.910. The assessment of the method with accuracy measure is depicted in figure 6c). For chunk size=3, the accuracy was evaluated by LSS-CNN, RNN, SLKNN+MLKNN, BPLion+LFNN, and proposed RIWO-based deep residual network are 0.821, 0.831, 0.842, 0.860, and 0.861. Likewise, for chunk size=6, the accuracy was evaluated by LSS-CNN, RNN, SLKNN+MLKNN, BPLion+LFNN, and proposed RIWO-based deep residual network is 0.824, 0.838, 0.849, 0.868, and 0.870. The performance improvement of LSS-CNN, RNN, SLKNN+MLKNN, BPLion+LFNN with respect to the proposed RIWO-based deep residual network considering accuracy are 5.287%, 3.678%, 2.413%, and 0.229%.

4.7. Comparative discussion

Table 3 presents an assessment of techniques with accuracy, TPR, TNR considering Reuter and 20 Newsgroup datasets. Considering the Reuter dataset with mapper=3, the highest accuracy of 0.830 is evaluated by developed RIWO-based deep residual network while the accuracy of existing LSS-CNN,

RNN, SLKNN+MLKNN, BPLion+LFNN are 0.800, 0.812, 0.818, and 0.819. The maximal TPR of 0.925 is measured by the proposed RIWO-based deep residual network, while the TPR is computed by LSS-CNN, RNN, SLKNN+MLKNN, BPLion+LFNN are 0.846, 0.850, 0.865, and 0.896. The highest TNR of 0.880 is computed by the proposed RIWO-based deep residual network, while the TPR of LSS-CNN, RNN, SLKNN+MLKNN, BPLion+LFNN are 0.824, 0.839, 0.849, and 0.863. With mapper=4, the highest TPR of 0.850, TNR of 0.940, and accuracy of 0.887 are evaluated by a developed RIWO-based deep residual network. With 20 Newsgroup datasets and mapper=3, the highest TPR of 0.840, highest TNR of 0.879, and highest accuracy of 0.859 are computed by the proposed RIWO-based deep residual network. With mapper=4, the highest TPR of 0.859, highest TNR of 0.910, and highest accuracy of 0.870 are evaluated by developed RIWO-based deep residual network.

Table 3. Comparative discussion

Datasets	Mappers	Metrics	LSS-CNN	RNN	SLKNN+MLKNN	BPLion+LFNN	Proposed RIWO-based deep residual network
Reuters dataset	Mapper=3	TPR	0.800	0.812	0.818	0.819	0.830
		TNR	0.846	0.850	0.865	0.896	0.925
		Accuracy	0.824	0.839	0.849	0.863	0.880
	Mapper=4	TPR	0.810	0.820	0.824	0.826	0.850
		TNR	0.855	0.856	0.876	0.900	0.940
		Accuracy	0.833	0.849	0.852	0.868	0.887
20 Newsgroup dataset	Mapper=3	TPR	0.796	0.815	0.818	0.829	0.840
		TNR	0.833	0.840	0.849	0.854	0.879
		Accuracy	0.824	0.830	0.839	0.856	0.859
	Mapper=4	TPR	0.810	0.827	0.836	0.849	0.859
		TNR	0.836	0.839	0.851	0.871	0.910
		Accuracy	0.824	0.838	0.849	0.868	0.870

5. Conclusion

A technique is presented for text classification in big data considering the MapReduce model. The purpose is to provide a hybrid optimization-driven deep learning model for text classification. Here, the pre-processing is carried out with stemming and stop word removal. In addition, the mining of significant features is performed wherein SentiWordNet features, contextual features, and thematic features are mined from input pre-processed data. Furthermore, the selection of the best features is carried out with Tanimoto similarity. The Tanimoto similarity examines the similarity between the features and selects the pertinent features with higher feature selection accuracy. Then, a deep residual network is employed for dynamic text classification. Here, the deep residual network is trained by the Adam algorithm. In addition, dynamic learning is carried out with the proposed RIWO-based deep residual network along with fuzzy theory for incremental text classification. Here, the training of the deep residual network is performed using the proposed RIWO. The proposed RIWO algorithm is the integration of IWO and ROA. The proposed RIWO-based deep residual network outperformed other techniques with the highest TPR of 85%, TNR of 94%, and accuracy of 88.7%. In the future, other datasets can be employed to validate the feasibility of the developed model.

References

- [1] Ranjan, N.M. and Prasad, R.S.,” LFNN: Lion fuzzy neural network-based evolutionary model for text classification using context and sense based features,” Applied Soft Computing, vol. 71, pp.994-1008, 2018.

- [2] Kotte, V.K., Rajavelu, S. and Rajsingh, E.B., “A similarity function for feature pattern clustering and high dimensional text document classification”, *Foundations of Science*, pp.1-18, 2019.
- [3] Wang, Jiaying; Li, Yaxin; Shan, Jing; Bao, Jinling; Zong, Chuanyu; Zhao, Liang "Large-Scale Text Classification Using Scope-Based Convolutional Neural Network: A Deep Learning Approach", *IEEE Access*, vol.7, pp.171548–171558, 2019.
- [4] Kuppli, Venkatanareshbabu; Biswas, Mainak; Edla, Damodar Reddy; Prasad, K. J. Ravi; Suri, Jasjit S. “A Mechanics-Based Similarity Measure for Text Classification in Machine Learning Paradigm”, *IEEE Transactions on Emerging Topics in Computational Intelligence*, pp. 1–21, 2018.
- [5] Liu, C. and Wang, X., “Quality-related English text classification based on recurrent neural network”, *Journal of Visual Communication and Image Representation*, vol.71, pp.102724, 2020.
- [6] Qi Li, Pengfei Li, MaoKezhi, Edmond Lo and Yat-Man, “Improving convolutional neural network for text classification by recursive data pruning”, *Neurocomputing*, vol. 414, pp. 143–152, 2020.
- [7] BenSaid, F. and Alimi, A.M., “Online feature selection system for big data classification based on multi-objective automated negotiation”, *Pattern Recognition*, vol. 110, pp.107629, 2020.
- [8] Jiang, M., Liang, Y., Feng, X., Fan, X., Pei, Z., Xue, Y. and Guan, R., “Text classification based on deep belief network and softmax regression”, *Neural Computing and Applications*, vol. 29, no. 1, pp.61-70, 2018.
- [9] T. W. Schoenharl and G. Madey, “Evaluation of Measurement Techniques for the Validation of Agent-Based Simulations Against Streaming Data”, Berlin, Germany: Springer, 2008.
- [10] C. G. Gonzalez, W. Bonventi, Jr., and A. L. Vieira Rodrigues, “Density of ‘ closed balls in real-valued and autometrizedboolean spaces for clustering applications,” In *Proceedings of 19th Brazilian Symposium of Artificial Intelligence*, pp. 8–22, 2008.
- [11] Kingma, D.P. and Ba, J., “Adam: A method for stochastic optimization”, 2014.
- [12] S. Tata and J. M. Patel, “Estimating the selectivity of tf-idf based cosinesimilarity predicates,” *ACMSigmod Rec.*, vol. 36, no. 2, pp. 7–12, 2007.
- [13] Siwei Lai, Liheng Xu, Kang Liu, and Jun Zhao, “Recurrent convolutional neural networks for text classification”, In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, pp. 2267–2273, 2015.
- [14] T. Wu, S. Liu, J. Zhang, and Y. Xiang, “Twitter spam detection based ondeep learning,” in *Proc. ACSW*, Art. no. 3, 2017.
- [15] M. V. Mäntylä, D. Graziotin, and M. Kuutla, “The evolution of sentiment analysis—A review of research topics, venues, and top cited papers”, *Computer Science Review*, vol. 27, pp. 16–32, Feb. 2018.
- [16] Junxian He, Zhiting Hu, Taylor Berg Kirkpatrick, Yink Huang and Eric P. Xing. “Efficient correlated topic modeling with topic embedding”, In *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 225-233, 2017.
- [17] Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, and Jeff Klingner, “Google's neural machine translation system: Bridging the gap between human and machine translation”, 2016.
- [18] Kim, J.K., Tur, G., Celikyilmaz, A., Cao, B. and Wang, Y.Y., “Intent detection using semantically enriched word embeddings”, In *2016 IEEE Spoken Language Technology Workshop (SLT)*, pp. 414-419, December 2016.
- [19] Chen, J., Yan, S. and Wong, K.C., “Verbal aggression detection on Twitter comments: Convolutional neural network for short-text sentiment analysis”, *Neural Computing and Applications*, pp.1-10, 2018.

- [20] Cheng, P. M., and Malhi, H. S., "Transfer learning with convolutional neural networks for classification of abdominal ultrasound images", *Journal of Digital Imaging*, vol. 30, no. 2, pp. 234-243, 2017.
- [21] Esteva, A., Kuprel, B., Novoa, R.A., Ko, J., Swetter, S.M., Blau, H.M. and Thrun, S., "Dermatologist-level classification of skin cancer with deep neural networks", *Nature*, vol. 542, no. 7639, pp. 115-118, 2017.
- [22] Luo, and Yuan., "Recurrent neural networks for classifying relations in clinical notes", *Journal of Biomedical Informatics*, vol. 72, pp.85-95, 2017.
- [23] Zhang, M., Li, W., and Du, Q., "Collaborative classification of hyperspectral and visible images with convolutional neural network", *Journal of Applied Remote Sensing*, vol. 11, no. 4, pp. 1, 2017.
- [24] 20 newgroup dataset extracted from, "<https://www.kaggle.com/crawford/20-news-groups>," accessed on November 2020.
- [25] Reuter dataset taken from, "<https://www.kaggle.com/nltkdata/reuters>," accessed on November 2020.
- [26] Chen, Z., Chen, Y., Wu, L., Cheng, S. and Lin, P., "Deep residual network-based fault detection and diagnosis of photovoltaic arrays using current-voltage curves and ambient conditions", *Energy Conversion and Management*, vol. 198, pp.111793, 2019.
- [27] Sang, H.Y., Duan, P.Y. and Li, J.Q., "An effective invasive weed optimization algorithm for scheduling semiconductor final testing problem", *Swarm and Evolutionary Computation*, vol. 38, pp.42-53, 2018.
- [28] Binu, D. and Kariyappa, B.S., "RideNN: A new rider optimization algorithm-based neural network for fault diagnosis in analog circuits" *IEEE Transactions on Instrumentation and Measurement*, vol. 68, no. 1, pp.2-26, 2018.
- [29] DHarsha Dave, and Shree jaswal; "Multiple text document summarization system using hybrid summarization technique", In *IEEE 1st International Conference on Next Generation Computing Technologies (NGCT)*, pp. 804-808, September 2015.
- [30] Oguzhan Tas, and Farzad Kiyani; "A survey automatic text summarization", *PressAcademia Procedia*, vol.5, no.1, pp.205-213, 2007.
- [31] Ghosh, M. and Kar, A., "Unsupervised linguistic approach for sentiment classification from online reviews using SentiWordNet 3.0," *Int J Eng Res Technol*, vol.2, no.9, 2013.
- [32] Vidyadhari Ch, Sandhya N, Premchand P., "A Semantic Word Processing Using Enhanced Cat Swarm Optimization Algorithm for Automatic Text Clustering", *Multimedia Research*, Vol.2, No.4, pp.23-32, 2019.
- [33] Abdalla, H. B., Ahmed, A. M., & Al Sibahee, M. A. (2020). Optimization Driven MapReduce Framework for Indexing and Retrieval of Big Data. *KSII Transactions on Internet and Information Systems (TIIS)*, 14(5), 1886-1908.
- [34] M. Mohsin, H. Li and H. B. Abdalla, "Optimization Driven Adam-Cuckoo Search-Based Deep Belief Network Classifier for Data Classification," in *IEEE Access*, vol. 8, pp. 105542-105560, 2020, doi: 10.1109/ACCESS.2020.2999865.